Graph Based reccommendation Systems: Comparison Analysis between Traditional Techniques and Graph Based Techniques

Final Project Report

Ragunwash Raj
ragunwash.raj@student-cs.fr
CentraleSupélec
Gif-sur-Yvette, France

Wanci He
wanci.he@student-cs.fr
CentraleSupélec
Gif-sur-Yvette, France

Greshma Babu
greshma.babu@student-cs.fr
CentraleSupélec
Gif-sur-Yvette, France

Vivian Koutromani
vivian.koutroumani@student-cs.fr
CentraleSupélec
Gif-sur-Yvette, France

ABSTRACT

Graph based recommendation systems are more and more prevalent in recent years due to their accuracy in prediction and also their essence in tracing the relations between items and users. In this project, we first generate candidates by the traditional method, collaborative filtering, both item based and popularity based. Then, we implement and compare 3 graph based methods: random walk, Node2Vec neural embedding, and bipartite graph. We evaluated the models by computing the average distance between the actual co-purchase graph and the candidates generated for each item. The biparitite graph model performs the best among the three models, attaining an average distance of 2.8.

1 INTRODUCTION - MOTIVATION

E-commerce platforms, and especially Amazon, offer a massive variety of products and information. In order to help the user to find out information about the product , recommendation systems where developed.

Recommendation systems create a similarity between the user and items and exploit the similarity between user/item to make recommendations. They include techniques and algorithms, able to suggest “relevant” items to users. The suggested items should be the most relevant possible item for the user, so that the user would be more likely to choose those items: Amazon products, news articles, Youtube videos etc.

Focusing on an Amazon products dataset, items are ranked according to their relevancy, and the most relevant ones are shown to the user. Our recommendation system should be able to determine the relevancy, based on historical data. If a customer has recently bought a coffee machine, then the system should start recommending coffee capsules and other products related to coffee machine.

2 PROBLEM DEFINITION

Recommendation systems’ traditional approaches are divided into two main categories: collaborative filtering and content-based systems and include:

- Popularity based systems: they recommend items purchased and highly rated by most users. It is not a personalized recommendation though.
- Classification model based: Focuses on the features of the user and uses a classification algorithm to decide whether the user is interested or not in the product.
- Content based recommendations: It is based on the information on the contents of the item rather than on the user opinions. The main idea is if the user likes an item then he or she will like the "other" similar item.
- Collaborative Filtering: Two types:1) User-based and 2) Item based. User based recommends products that a user with similar taste have rated high. Item based assumes that a user likes similar products.
- Hybrid Approaches: This system approach is to combine collaborative filtering, content-based filtering, and other approaches .
- Association rule mining : Association rules capture the relationships between items based on their patterns of co-occurrence across transactions.

Traditional algorithms based on collaborative filtering require an up-to-date dataset of users and their preferences, which is difficult to gather for huge database of items. Also, content-based approach suffers from the complex computation of similarity among items. More recently, neural embedding, especially word2vec, has shown effective in the recommendation system. Over the last few years, Graph Learning based Recommendation Systems have been developed on graphs where the important objects, e.g., customers, products, and attributes, are either explicitly or implicitly connected.

A graph-based recommendation system using neural embedding to capture similarity in a sparse dataset, seem sensible and thus worth exploring. Given the Amazon product co-purchasing network metadata, we use neural embedding to predict the co-purchase bipartite graph of items and baskets of customers. We chose to work on this project because we think that recommendation systems are gaining popularity among platforms, now that web is the dominant type of advertising. We consider it as an extremely interesting application of network science and hence, would like to explore and experiment with it.

3 RELATED WORK

Many scientists have focused their research objective in forming graph based techniques for reccommendation systems.

A very common approach among such systems is to perform graph traversal techniques on a basket-to-item bipartite graph and generate a set of co-purchased candidates. Li, in his paper "Recommendation as link prediction in bipartite graphs" [2], suggests a kernel-based recommendation approach that examine user-item pairs, to predict whether there is an edge between customers and items. He used a set of users and a set of items as nodes, and each transaction (purchase of a product) was represented as an an edge between the two. The graph kernel is defined on the user-item pairs context, so predicting a link between user-item, based on graph analysis, he addresses the item recommendation problem.

Mikolov et al, in his paper 'Efficient Estimation of Word Representations in Vector Space'[3], examine and evaluate a technique to embed natural language words in an n-dimensional vector space (the word2vec model). In the word2vec model, each word is defined by its context words that occur within a certain proximity in sentences. He trains a neural-network on n-gram tuples generated with each word and its n-gram context words so that the the context word through the network result in the target word. To achieve this CBOW (Continuous Bag of Words) model and the Skip-gram model,several tecniques are used. Although the two methods differ in the way the neural network is trained, they both use the neural network as an autoencoder in which one of its layers, the weight matrix, represents a concatenation of vector representation of the natural language vocabulary. More importantly, Mikolov's paper explores the capability of capturing word analogies. Certain relational properties between natural language words can be described as vector operations, such as (Men-Women) - (King-Queen). The paper got considerable attention over the years, and has been used for Recommendation Systems as well.

Ivan F. Videla-Cavieres and Sebastian A. Rios have also in their paper "A real case"[4], explore traditional methods of getting information from data in retail, like the market basket analysis, which is one of the most applied techniques over transactional data to discover frequent itemsets, clustering techniques such as K-means and SOM. Their results in real supermarket data are of poor quality. So, they explored a new approach of performing MBA based on overlapping communities using graph mining techniques. They generated a network of products, based only on transactions, where each product is linked to others when they appear in the same basket from the same buyer, creating a co-purchased product network. They first built a set of temporal information using transaction set, followed by the transactional product bipartite network where each transaction is linked to the products that are purchased in that particular transaction. Next, they created a co-purchased product network from the bipartite network, where the nodes represent the products and the edges represent the number transactions between the two. Finally, they utilized the overlapping community detection to identify the strongly connected nodes, or clusters.

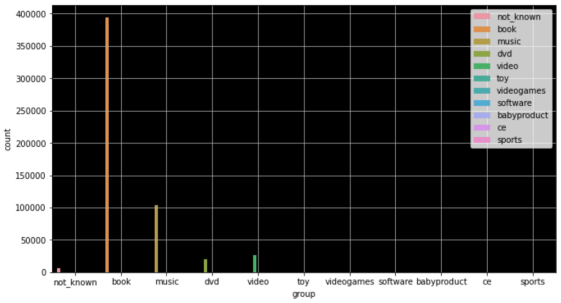

Figure 1: Different categories in the Amazon dataset

This method, resulted in much better clusters, hence more insightful recommendation

4 METHODOLOGY

For our project, we explored multiple publicly available data sources, finally choosing a dataset from Amazon, containing products and their reviews as well as copurchased products, and focusing on only music and books as that was the larger chunk of the data (see Figure 1).

4.1 Data Preprocessing and Exploration

In order to get the data into usable format we created a dictionary where key is ASIN which is the target product that is bought and value is again a dictionary holding different attributes for the target product for ex, co purchased items, ratings list, user-list, categories etc. The data description is presented in Table 1. Since different algorithm required different format for the data as in CF required data in a rating matrix form while for random walk and other model we need data in Graphical structures. also there were around 8000 rows which has not received any reviews from the user, considering the memory usage and keeping in mind the the importance of graph edge weights we decided to drop them.

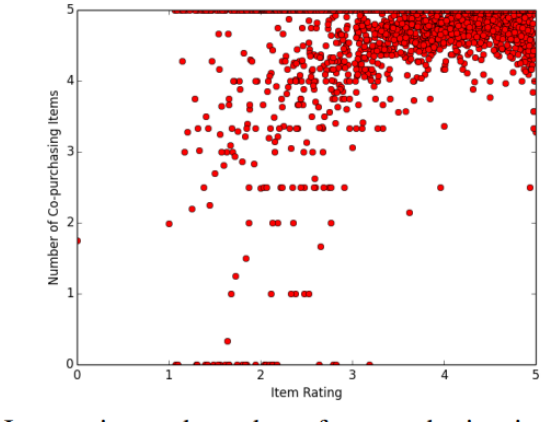

Figure 2: Item rating analysis

The first step we carried out was to understand the data. We carried out some basic analysis like how many products and users

Table 1: Data Description

Object	Number
Total number of products purchased	548552
Total number of users	1555170
Total number of product with at least one review	402724
Total number of co-purchased products	410057
Total number of products	721342
Total number of pairs of products	1788725
Average number of co-purchasing products per product	3.2600
Total number of review	7593244
Average number of reviews per product	13.8423
Average number of ratings per product	3.1752

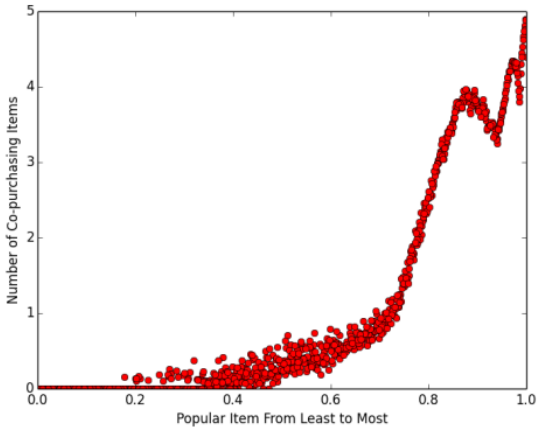

Figure 3: Popularity vs Copurchasing analysis

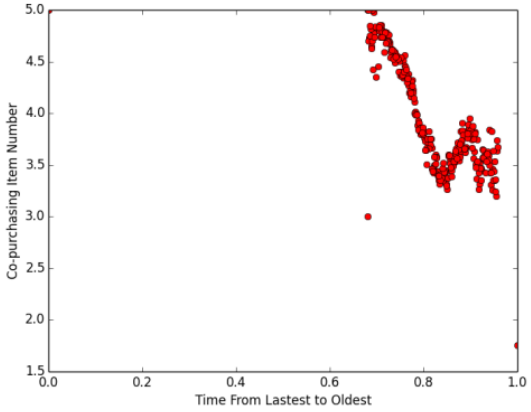

Figure 4: Time of first review analysis

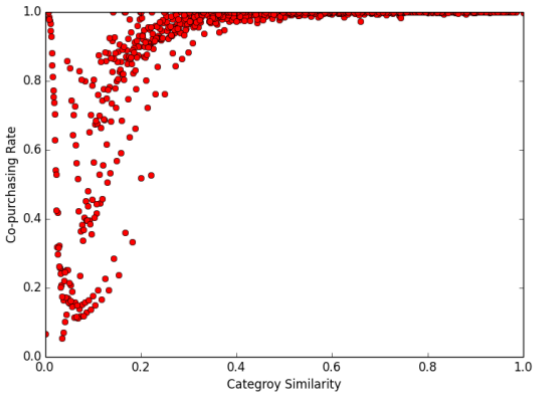

Figure 5: Similarity of category vs co-purchasing

were in the data, the average number of reviews and ratings, which we have detailed below.

We then tried to study how and whether the item ratings and co-purchasing are related (see Figure 2), i.e. if there is a correlation between the average rating of a product and the number of co-purchasing products the product has. We see that generally, the higher ranked products have more co-purchased products.

We also performed sales rank analysis to see if there is a correlation between the popularity of a product and the number of co-purchasing products the product has (see Figure 3).

Another analysis we carried out was based on the timing of the review. The time of first review analysis tries to see if there is a correlation between the time the product is first reviewed and the number of co-purchasing products the product has (see Figure 4).

Furthermore, we also analyzed whether being similar in titles of categories caused products to be co-purchased more often (see Figure 5). We found that having similar catgories did have a positive impact on being co-purchased. In fact, category similarity had a correlation: of 0.56355022 with being co-purchased.

As both the music and book datasets were massive, we used samples to train and test our algorithms. On the samples, again we filtered out products that have less than 50 reviews, and users that have reviewed less than 50 products. For both datasets, number of unique products were much less than number of unique users, so item-based collaborative filtering makes much more sense. Our exploratory analysis showed that ratings were favorable in their

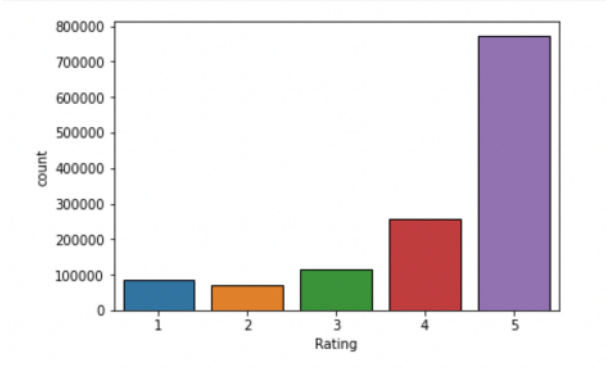

Figure 6: Count of ratings

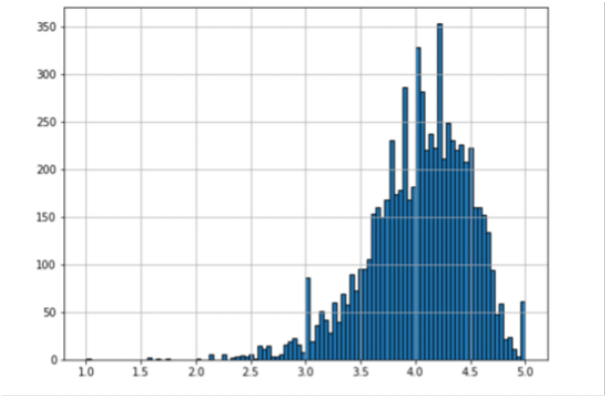

Figure 7: Book reviews distributions

majority (see Figure 6).

And that we had some outliers, users that rated more that 1,000 products, but the median of number of ratings was 14. Same applies for products, there were some outliers with many reviews(400 for books, 120 for music), but the median was much lower (25 reviews per book, 10 for music products).
For book reviews, we can see that the ratings' distribution is concentrated around 4 stars, while for music products it's more balanced, but still skewed.

4.2 Collaborative filtering - Item based

Amazon uses currently item-item collabrative filtering, to provide recommendations to users, which scales to massive datasets and produces high quality recommendation system in the real time. It aims to fill in the missing entries of a user-item association matrix. CF is based on the idea that the best recommendations come from people who have similar tastes. In other words, it uses historical item ratings of like-minded people to predict how someone would rate an item. In our approach we used KNNWithMeans algorithm, which is a basic collaborative filtering algorithm, taking into account the mean ratings of each user, taking into account 5 neighbors for aggregation and pearson similarity, which is the covariance of the two n-dimensional vectors divided by the product of their standard deviations. This model gave us an RMSE of 1.66 for the music

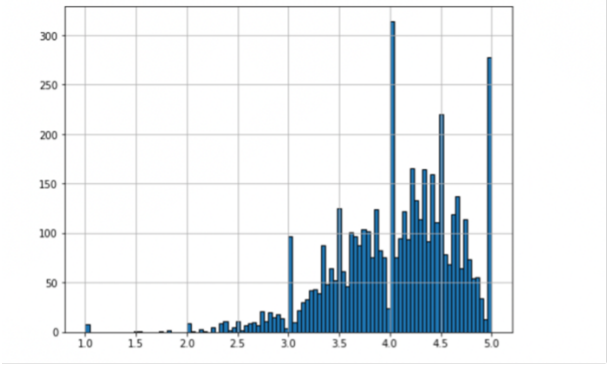

Figure 8: Music reviews distributions

test set, and 0.77 on the book test set. For this purpose, we also tried the Turi Create's recommender (TuriCreate is an open-source python library, used to create core machine learning models for supervised and unsupervised learning), but the RMSE we got was much worse than ours.

4.3 Popularity-Based

Popularity based recommendation system works with trends. It basically uses the items which are in trend right now. The problem with popularity based recommendation system is that it's not personalized so we don't take advantage of the individuals' behaviors. In our approach, first we filtered out users that have given less than 50 reviews. For the remaining reviews, we counter the sum of ratings for each product, and the number of reviews per product, which works as a score to pick the top 5 recommendations. Comparing the recommendations' average rating, with the actual ratings user have given to these products, we got an RMSE of 1.104 for books, and 1,217 for music.

4.4 Matrix Factorization

Matrix factorization is a collaborative filtering method which focuses on the relationship between items and users. Latent features, the association between users and items matrices, are determined to find similarity and make a prediction based on both item and user entities. With the input of users' ratings on the shop items, we would like to predict how the users would rate the items so the users can get the recommendation based on the prediction. Given a ranking matrix of users and items, we want to fill the "gaps" The matrix factorization of user and item matrices can be generated when the math cost function RMSE is minimized through matrix factorization. Gradient descent is a method to minimize the cost function. We used the TuriCreate Factorization Reccommender, that took into account 8 factors of the books (and music respectively) and optimized the recommendations using the Stohastic Gradient Descent to minimize RMSE. The final RMSE on the test set reccommendations was 0.91

4.5 Market Basket Analysis

In market basket analysis, we consider different market baskets and the products purchased. We analyze the products pairwise (or in groups of three or more) to see in how many baskets they co-occur versus how many baskets they individually appear in to see if the co-occurrence is more because the product itself is purchased a lot or if the co-occurrence is due to some inherent the consumer

buying trend in which case we can formulate association rules, which are basically if-then statements linking multiple products. Here, we use the co-purchased lists to analyze how many products co-occur and how often they occur to obtain the association rules.

4.6 Random Walk

To compare our recommendation systems, as a baseline, we also made use of a random walk system. The idea behind the random walk is this: if we consider a graph neural network with the items as nodes connected to each other if they are co-purchased, if we perform a random walk from two nodes, the chances of them overlapping would be more if the two nodes are similar. Using these idea, we considered random walks of 10 steps from each node (with a teleportation probability of 0.4 back to itself to ensure that the final destination of the random walk is not too random) and computed the Jaccard similarity of the final destinations of 10 such random walks. Our random walk recommendation system would recommend items that have higher Jaccard similarity.

4.7 Recommendation using Neural Embeddings

Neural embeddings have shown to be very effective in generating a representation of items in high dimensional latent space. In this paper, we use the implementation of DeepWalk provided in node2vec. DeepWalk learns *d*-dimensional feature representations by simulating uniform random walks[1]. As per DeepWalk, node2vec takes the latent embedding of the walks, with negative sampling, and uses them as input to a neural network to classify nodes. In this way, we learn embeddings using Gensim Word2Vec by optimizing the training algorithm, skipgram, using Stochastic Gradient Descent. We output an embedding with the shape of (50,) for each item and compute the cosine similarity between each pair of items for recommending similar items.

4.8 Recommendation using Bipartitte Graph

A bipartite graph is a graph model composed of two groups of nodes with different properties, and the nodes in the same group are not connected. A bipartite graph can be defined as network structure G = <U, I, E> where U denotes the user set, I denotes the item set, and E denotes the edges of bipartite graph model. we took only those co products in the performance for which the information has been provided ,we created a bipartite graph in which a pair of node is basically the item and its corresponding co purchased item and edge exist between them whose weight are computed using two different strategy method 1: for the source node and the target node we compute the element wise expectation between useful votes and the rating and obtained a score between 0 and 1

rating[i]*(normalisevote+normalisehelpfulvote)/totalrating

in order to get the weight we multiply the score of target node and source node. method2: we computed the text similarity between the category of the source product and the target product. the edge weights in the graph is a measure of the similarity between the books connected by the edge. So we can use the island method to only retain edges with threshold >= 0.5, and assign resulting graph to purchasedAsinEgoTrimGraph.Now given the purchasedAsinEgoTrimGraph we constructed above, we can get all the list of nodes connected to the purchasedAsin beyond certain threshold. we Find the list of neighbors of the purchasedAsin in the purchasedAsinEgoTrimGraph, and assign it to purchasedAsinNeighbors,then we pick the Top Five book recommendations from among the purchasedAsinNeighbors based on one or more of the following data of the neighboring nodes: SalesRank, AvgRating, TotalReviews, DegreeCentrality, and ClusteringCoeff, For our case we picked books based on product of average rating and total reviews.

Table 2: Performance Comparison

Models	Average Distance
RandomWalk	1.9453
Neural Embedding	2.3387
Bipartite Graph	2.8

5 EVALUATION

After attaining the Top 5 recommended candidates from each model, we compute the average distance between the target nodes contained in the real copurchase graph and the candidates we generated. The distance is used as our performance evaluation metric, which is presented as in Table 2.

6 CONCLUSIONS

we looks at model of the recommendation system, such as Collaborative Filtering, Random-Walk algorithms,neural embedding candidate generation and recommendation using bipartite copurchasegraphs.We observed that traditional methods like CF failed to get good similarity in bigger database ,the memory consumption is very high and computation time is exponential as well. The performance also drops for bigger rating matrix's because of the huge sparsity. Graph model as very efficient in such cases as they are able to capture the embedding information more effectively.Based on the results of our data analysis and prediction of different combinations of factors, we find that text similarity is really helpful when predicting joint purchasing pairs, meaning that items within the same categories are often purchased together more often. We feel that the recommendation performance can be further improved if we used a Graph Neural network to get the embedding for the given product as they are able to capture direct and hidden similarity based on the structure of the graph and hence we can compute a more robust similarity between the item-item pair.

REFERENCES

[1] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[2] Xin Li and Hsinchun Chen. 2013. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems* 54, 2 (2013), 880–890.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[4] Ivan F Videla-Cavieres and Sebastián A Ríos. 2014. Extending market basket analysis with graph mining techniques: A real case. *Expert Systems with Applications* 41, 4 (2014), 1928–1936.